EECS 349: Machine Learning

Bryan Pardo

Topic: Concept Learning and Version Spaces

(with some tweaks by Doug Downey)

Concept Learning

- Much of learning involves acquiring general concepts from specific training examples
- *Concept*: subset of objects from some space
- Concept learning: Defining a function that specifies which elements are in the concept set.

A Concept

- Let there be a set of objects, ***X.***
  ***X* = {**White Fang, Scooby Doo, Wile E, Lassie**}**

- *A concept* **C** is…

  A subset of ***X***

  C = dogs = {Lassie, Scooby Doo**}**

  A function that returns ***1*** only for elements in the concept

  C(Lassie) = 1, **C**(Wile E) = 0

Instance Representation

- Represent an object (or *instance*) as an *n*-tuple of *attributes*
- Example: Days (6-tuples)

Sky	**Temp**	**Humid**	**Wind**	**Water**	**Forecast**
sunny	warm	normal	strong	warm	same
sunny	warm	high	strong	warm	same
rainy	cold	high	strong	warm	change
sunny	warm	high	strong	cool	change

Example Concept Function

- "Days on which my friend Aldo enjoys his favorite water sport"

$$\overbrace{\hspace{20em}}^{\text{INPUT}}$$

OUTPUT

Sky	Temp	Humid	Wind	Water	Forecast	C(x)
sunny	warm	normal	strong	warm	same	1
sunny	warm	high	strong	warm	same	1
rainy	cold	high	strong	warm	change	0
sunny	warm	high	strong	cool	change	1

Hypothesis Spaces

- Hypothesis Space ***H***: subset of all possible concepts

- For learning, we restrict ourselves to ***H***

  H may be only a *small subset* of all possible concepts (this turns out to be important – more later)

Example: MC2 Hypothesis Space

- MC2 (Mitchell, Chapter 2) hypothesis space
  - Hypothesis *h* is a conjunction of constraints on attributes
- Each constraint can be:
  - A specific value : e.g. *Water=Warm*
  - A don't care value : e.g. *Water=?*
  - No value allowed: e.g. *Water=Ø*
- Instances x that satisfy the constraints of *h* have h(x) = 1, otherwise h(x) = 0
- Example hypotheses:

Sky	**Temp**	**Humid**	**Wind**	**Water**	**Forecast**
sunny	?	?	?	?	?
?	warm	?	?	?	same

Concept Learning Task

GIVEN:

- Instances X
  - E.g., days decribed by attributes:
    Sky, Temp, Humidity, Wind, Water, Forecast

- Target function c:
  E.g., EnjoySport X $\rightarrow$ {0,1}

- Hypothesis space H
  - E.g. MC2, conjunction of literals: < Sunny ? ? Strong ? Same >

- Training examples D
  - positive and negative examples of the target function: $<x_1,c(x_1)>,\ldots, <x_n,c(x_n)>$

FIND:

- A hypothesis h in H such that h(x)=c(x) for all x in D.

Concept Learning Task

GIVEN:

- Instances X
  - E.g., days decribed by attributes:
    Sky, Temp, Humidity, Wind, Water, Forecast

- Target function c:
  E.g., EnjoySport X → {0,1}

- Hypothesis space H
  - E.g. MC2, conjunction of literals: < Sunny ? ? Strong ? Same >

- Training examples D
  - positive and negative examples of the target function: $<x_1,c(x_1)>,\ldots,<x_n,c(x_n)>$

FIND:

- A hypothesis h in H such that h(x)=c(x) for all x in D.

Inductive Learning Hypothesis

- Any hypothesis found to approximate the target function well over the training examples, will also approximate the target function well over the unobserved examples.

- This might not be true. When it it isn’t the hypothesis does not generalize well.

Number of Instances, Concepts, Hypotheses

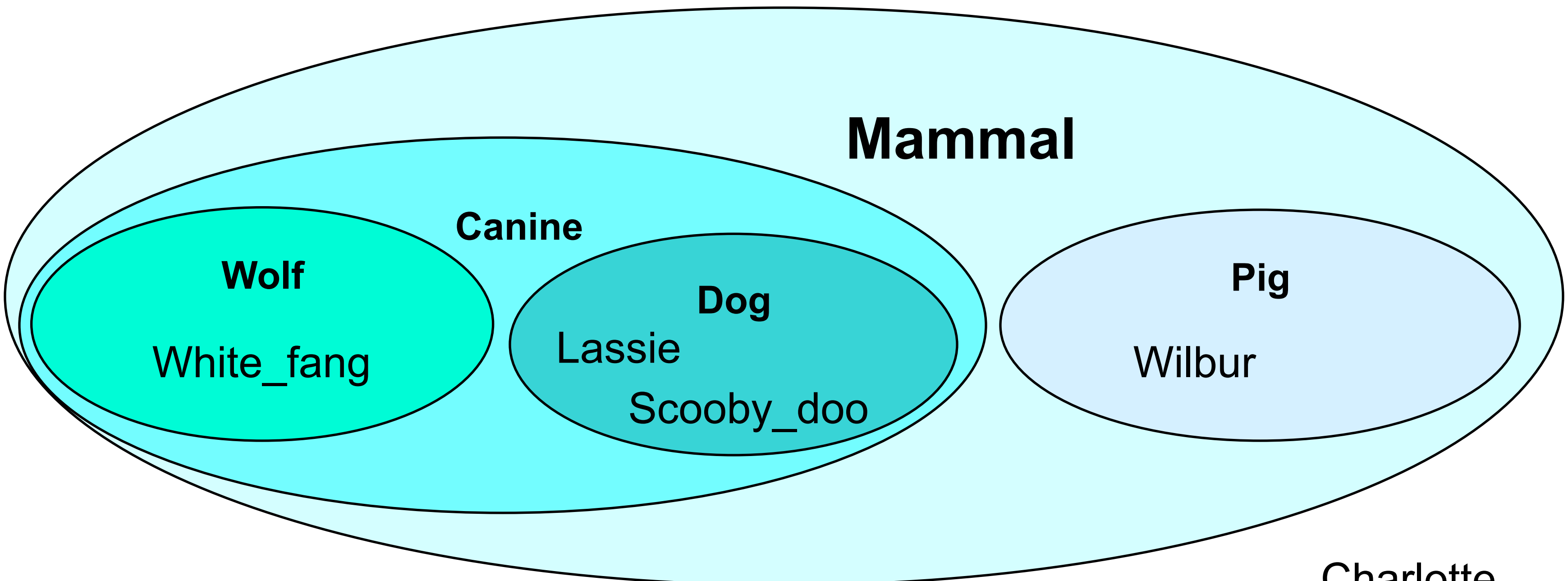
- Sky: Sunny, Cloudy, Rainy
- AirTemp: Warm, Cold
- Humidity: Normal, High
- Wind: Strong, Weak
- Water: Warm, Cold
- Forecast: Same, Change

distinct instances : $3*2*2*2*2*2 = 96$
distinct concepts: 2^{96}

syntactically distinct hypotheses in MC2: $5*4*4*4*4*4=5120$
semantically distinct hypotheses in MC2: $1+4*3*3*3*3*3=973$
Number of possible hypothesis spaces: $2^{2^{96}}$

Concept Generality

- A concept P is **more general than or equal to** another concept Q iff the set of instances represented by P includes the set of instances represented by Q.



Charlotte

General to Specific Order

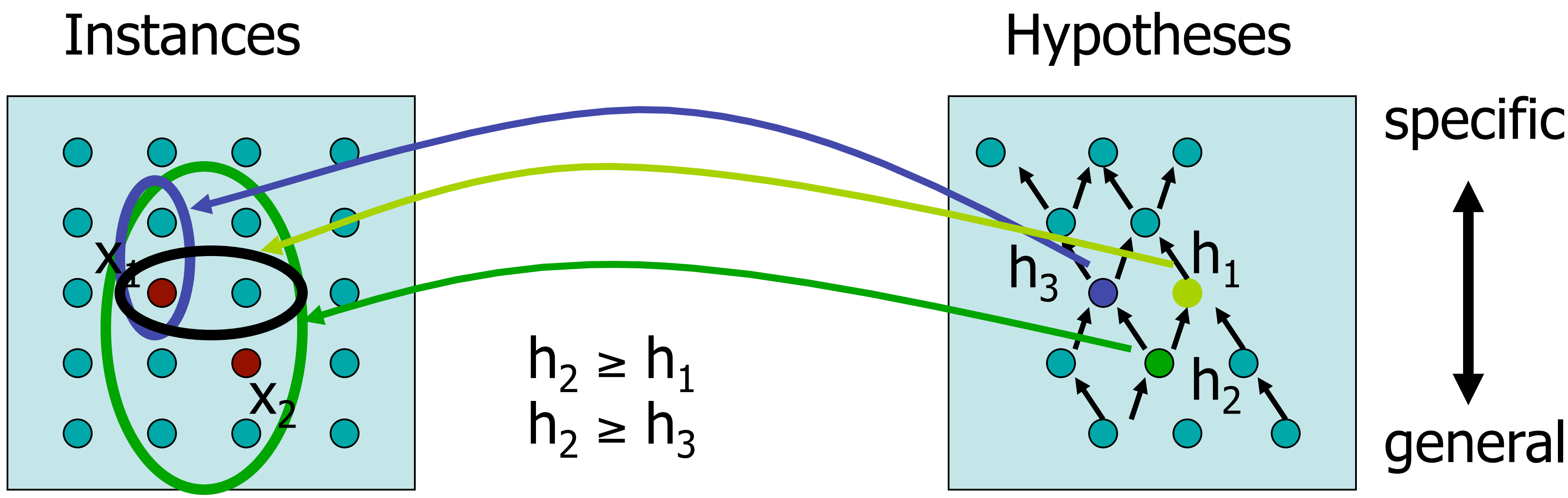
- Consider two hypotheses:
  - h_1=< Sunny,?,?,Strong,?,?>
  - h_2=< Sunny,?,?,?,?,?>

- Definition: h_j is **more general than or equal to** h_k iff:

$$h_j \geq h_k \equiv \forall x \left(h_k(x) = 1 \rightarrow h_j(x) = 1 \right)$$

- This imposes a partial order on a hypothesis space.

Instance, Hypotheses and ”generality”



x_1=< Sunny,Warm,High,Strong,Cool,Same>

x_2=< Sunny,Warm,High,Light,Warm,Same>

h_1=< Sunny,?,?,Strong,?,?>

h_2=< Sunny,?,?,?,?,?>

h_3=< Sunny,?,?,?,Cool,?>

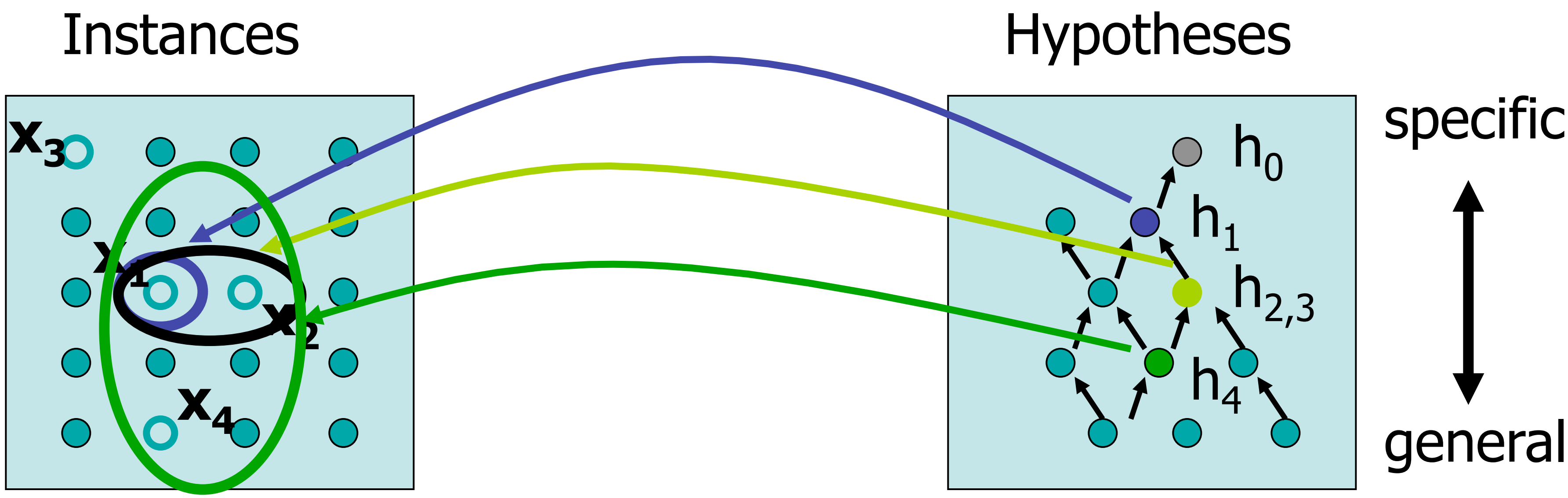
Find-S Algorithm

1. Initialize ***h*** to the most specific hypothesis in **H**

2. For each positive training instance ***x***:
   – For each attribute constraint $\boldsymbol{a_i}$ in ***h***
     - If the constraint is satisfied by ***x***, do nothing
     - else replace $\boldsymbol{a_i}$ in ***h*** by the next more general constraint that is satisfied by ***x***

3. Output hypothesis ***h***

Hypothesis Space Search by Find-S



x_1=<Sunny,Warm,Normal,Strong,Warm,Same>+

x_2=<Sunny,Warm,High,Strong,Warm,Same>+

x_3=<Rainy,Cold,High,Strong,Warm,Change> -

x_4=<Sunny,Warm,High,Strong,Cool,Change> +

h_0=< Ø, Ø, Ø, Ø, Ø, Ø,>

h_1=< Sunny,Warm,Normal, Strong,Warm,Same>

$h_{2,3}$=< Sunny,Warm,?, Strong,Warm,Same>

h_4=< Sunny,Warm,?, Strong,?,?>

Properties of Find-S

- When hypothesis space described by constraints on attributes (e.g., MC2)
  - Find-S will output the most specific hypothesis within H that is consistent with the positve training examples

Complaints about Find-S

- Ignores negative training examples
- Why prefer the most specific hypothesis?
- Can't tell if the learner has converged to the target concept, in the sense that it is unable to determine whether it has found the *only* hypothesis consistent with the training examples.

Version Spaces

- Hypothesis **h** is consistent with a set of training examples **D** of the target concept **c** iff **h(x)=c(x)** for each training example **<x,c(x)>** in **D**.

$$Consistent(h,D) \equiv \left(\forall \langle x, c(x)\rangle \in D\right) h(x) = c(x)$$

- A **version space** : all the hypotheses that are consistent with the training examples.

$$VS_{H,D} \equiv \{h \in H \mid Consistent(h,D)\}$$

- Imposing a partial order (like the ≥ one) on the version space lets us learn concepts in an organized way.

List-Then Eliminate Algorithm

1. *VersionSpace* $\leftarrow$ a list containing every hypothesis in H

2. For each training example <x,c(x)>

   remove from *VersionSpace* any hypothesis that is inconsistent with the training example $h(x) \neq c(x)$

3. Output the list of hypotheses in *VersionSpace*

Example Version Space

S: {<Sunny,Warm,?,Strong,?,?>}

<Sunny,?,?,Strong,?,?> <Sunny,Warm,?,?,?,?> <?,Warm,?,Strong,?,?>

G: {<Sunny,?,?,?,?,?>, <?,Warm,?,?,?>, }

x_1 = <Sunny Warm Normal Strong Warm Same> +
x_2 = <Sunny Warm High Strong Warm Same> +
x_3 = <Rainy Cold High Strong Warm Change> -
x_4 = <Sunny Warm High Strong Cool Change> +

Representing Version Spaces

- The **general boundary**, G, of version space $VS_{H,D}$ is the set of maximally general members.
- The **specific boundary**, S, of version space $VS_{H,D}$ is the set of maximally specific members.
- Every member of the version space lies between these boundaries

$$VS_{H,D} \equiv \{h \in H \mid \exists(s \in S), \exists(g \in G)\ g \geq h \geq s\}$$

Candidate Elimination Algorithm

Initialize G to the set of maximally general hypotheses in H
Initialize S to the set of maximally specific hypotheses in H
For each training example d, do

- If d is a positive example
  - Remove from G any hypothesis inconsistent with d
  - For each hypothesis s in S that is not consistent with d
    - Remove s from S
    - Add to S all minimal generalizations h of s such that
      - h is consistent with d, and some member of G is more general than h
    - Remove from S any hypothesis that is more general than another hypothesis in S
- If d is a negative example
  - Remove from S any hypothesis inconsistent with d
  - For each hypothesis g in G that is not consistent with d
    - Remove g from G
    - Add to G all minimal specializations h of g such that
      - h is consistent with d, and some member of S is more specific than h
    - Remove from G any hypothesis that is less general than another hypothesis in G

Candidate-Elimination Algorithm

- When does this halt?
- If S and G are both singleton sets, then:
  - if they are identical, output value and halt.
  - if they are different, the training cases were inconsistent. Output this and halt.
- Else continue accepting new training examples.

Example Candidate Elimination

S: {<∅, ∅, ∅, ∅, ∅, ∅ >}

G: {<?, ?, ?, ?, ?, ?>}

x_1 = <Sunny Warm Normal Strong Warm Same> +

S: {< Sunny Warm Normal Strong Warm Same >}

G: {<?, ?, ?, ?, ?, ?>}

x_2 = <Sunny Warm High Strong Warm Same> +

S: {< Sunny Warm ? Strong Warm Same >}

G: {<?, ?, ?, ?, ?, ?>}

Example Candidate Elimination

S: {< Sunny Warm ? Strong Warm Same >}

G: ~~{<?, ?, ?, ?, ?, ?>}~~

x_3 = <Rainy Cold High Strong Warm Change> -

S: {< Sunny Warm ~~?~~ Strong Warm Same >}

G: {<Sunny,?,?,?,?,?>, <?,Warm,?,?,?,?>, ~~<?,?,?,?,?,Same>~~}

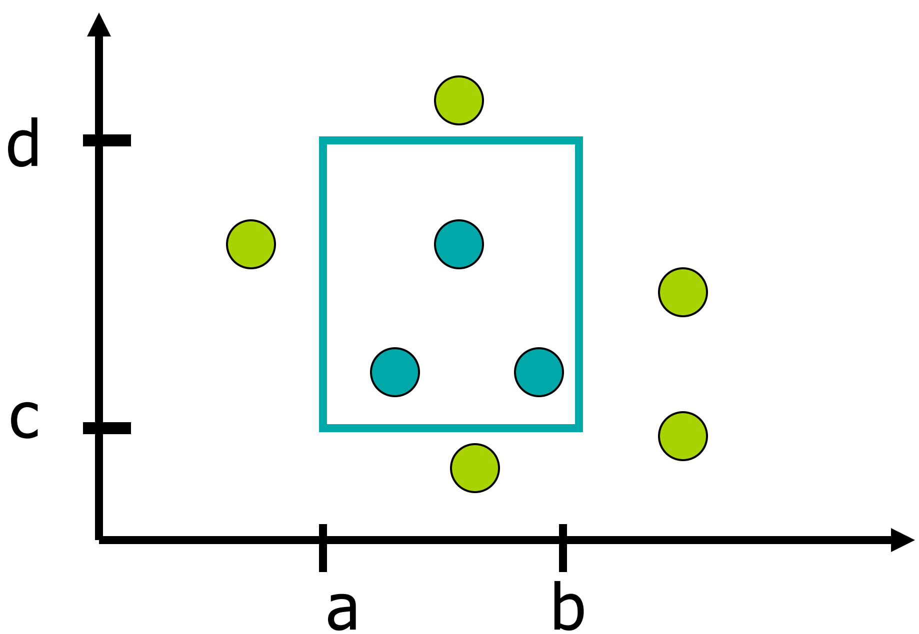
x_4 = <Sunny Warm High Strong Cool Change> +

S: {< Sunny Warm ? Strong ? ? >}

G: {<Sunny,?,?,?,?,?>, <?,Warm,?,?,?> }

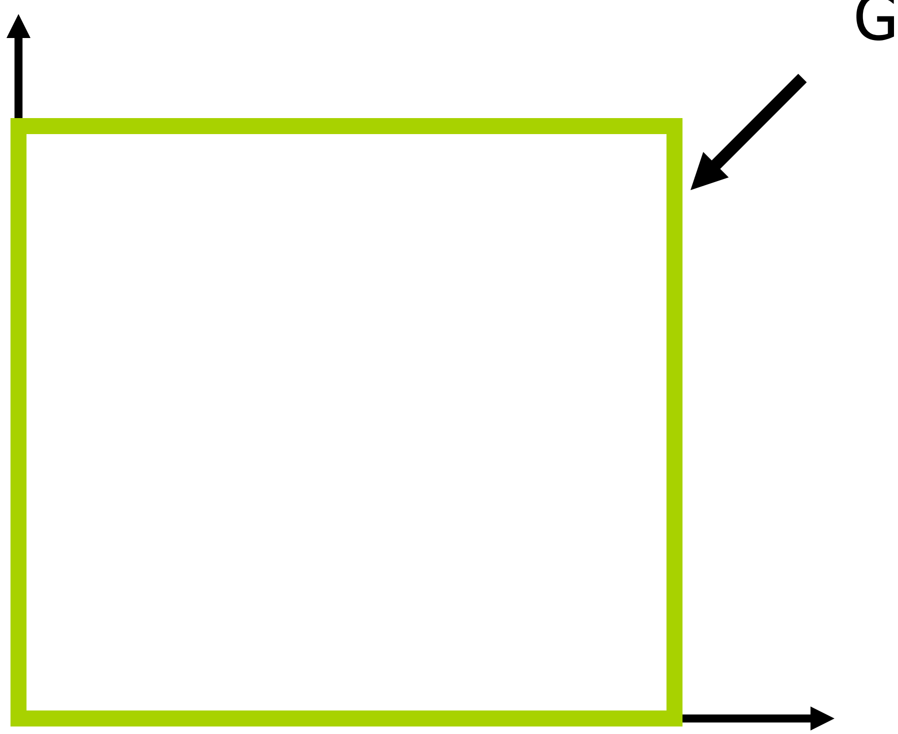
Example Candidate Elimination

- Instance space: integer points in the x,y plane with $0 \leq x,y \leq 10$
- hypothesis space : rectangles. That means hypotheses are of the form $a \leq x \leq b$, $c \leq y \leq d$



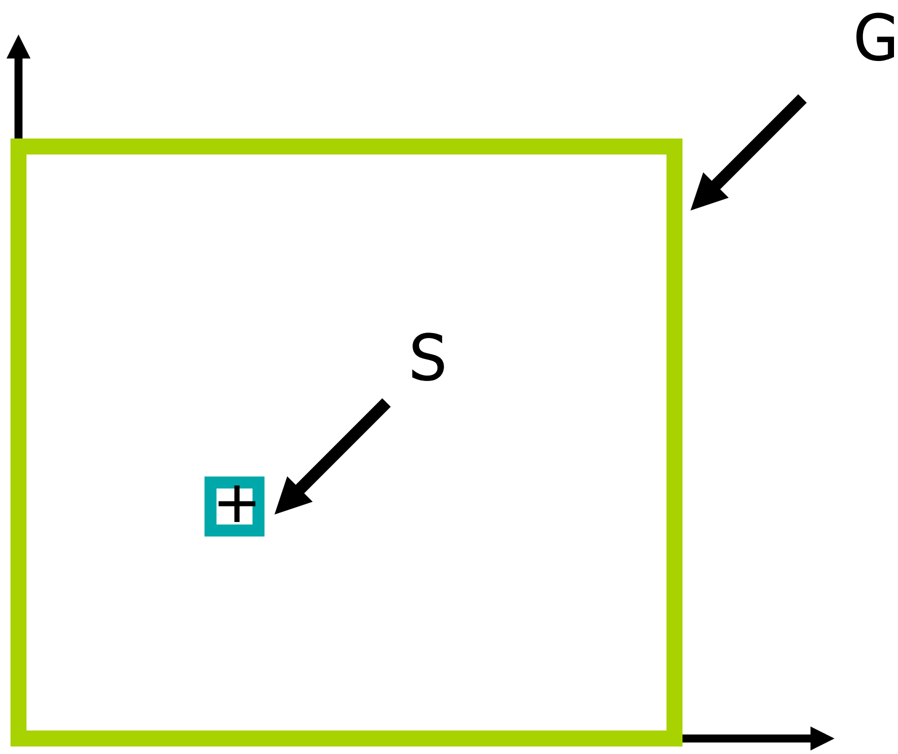
Example Candidate Elimination

- examples = {Ø}
- G={0,10,0,10}
- S={Ø}



Example Candidate Elimination

- examples = {(3,4),+}
- G={(0,10,0,10)}
- S={(3,3,4,4)}



Classification of Unseen Data

S: {<Sunny,Warm,?,Strong,?,?>}

<Sunny,?,?,Strong,?,?> <Sunny,Warm,?,?,?,?> <?,Warm,?,Strong,?,?>

G: {<Sunny,?,?,?,?,?>, <?,Warm,?,?,?>, }

x_5 = <Sunny Warm Normal Strong Cool Change> + 6/0
x_6 = <Rainy Cold Normal Light Warm Same> - 0/6
x_7 = <Sunny Warm Normal Light Warm Same> ? 3/3
x_8 = <Sunny Cold Normal Strong Warm Same> ? 2/4

Deductive reasoning

- Tries to show a conclusion MUST follow from a set of premises (axioms)
- What we typically think of as "Logic" (1^{st} order, 2^{nd} order, etc.)
- Covered in EECS 348.
- Example
  - All men are mortal
  - Socrates is a man
  - Therefore, Socrates is mortal

Inductive reasoning

- The premises of an inductive argument indicate support (often probabilistic support) but do not ensure the conclusions are true.
- Example
  - 93% of students are right-handed.
  - Will is a student.
  - Therefore, Will is right-handed.

Inductive Bias

- NOT the same as bias in a statistical estimator
- DEFINITION: The set of axioms that would need to be added to the knowledge of the system so that a deductive reasoner would make the same inference as the inductive reasoner.
  - Example: Will does whatever the majority does.

Inductive Bias

CHOICE OF PERFORMANCE MEASURE IS A KIND OF BIAS:

Means squared error (linear regression)

Maximum margin between classes (SVM)

Fitness function (Genetic algorithm)

Inductive Leap

+ <Sunny Warm Normal Strong Cool Change>
+ <Sunny Warm Normal Light Warm Same>

S : <Sunny Warm Normal ? ? ?>

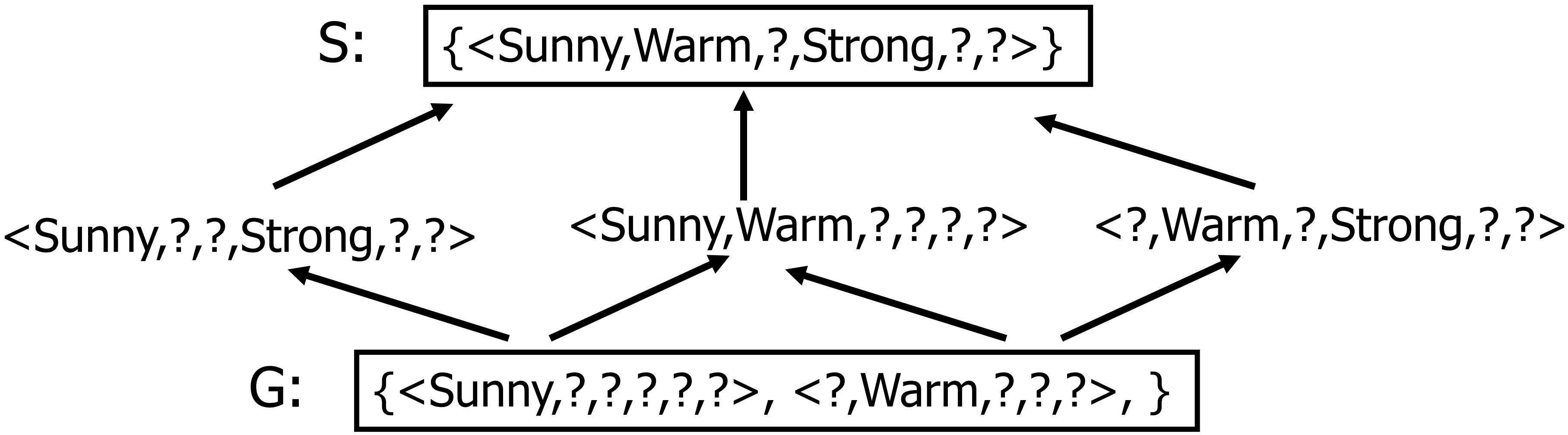
new example <Sunny Warm Normal Strong Warm Same>

How can we justify classifying the new example as positive?

Since S is the specific boundary all other hypotheses in the version space are more general. So if the example satifies S it will also satisfy every other hypothesis in VS.

inductive bias: Concept **c** can be described by a conjunction of literals.

What Example to Query Next?



- What would be a good query for the learner to pose at this point?
- Choose an instance that is classified positive by some of the hypotheses and negative by the others.

  <Sunny, Warm, Normal, Light, Warm, Same>
- If the example is positive S can be generalized, if it is negative G can be specialized.

Biased Hypothesis Space

- Our hypothesis space is unable to represent a simple disjunctive target concept :

(Sky=Sunny) v (Sky=Cloudy)

x_1 = <Sunny Warm Normal Strong Cool Change> +
S_1 : { <Sunny, Warm, Normal, Strong, Cool, Change> }

x_2 = <Cloudy Warm Normal Strong Cool Change> +
S_2 : { <?, Warm, Normal, Strong, Cool, Change> }

x_3 = <Rainy Warm Normal Strong Cool Change> -
S_3 : {} The third example x_3 contradicts the already overly general hypothesis space specific boundary S_2.

Unbiased Learner

- Idea: Choose H that expresses every teachable concept, that means H is the set of all possible subsets of X
- $|X|=96 \Rightarrow |H|=2^{96} \sim 10^{28}$ distinct concepts
- H = disjunctions, conjunctions, negations
  <Sunny Warm Normal ? ? ?> v <? ? ? ? ? Change>
- H surely contains the target concept.

Unbiased Learner

Assume positive examples (x_1, x_2, x_3) and negative examples (x_4, x_5)

$$S : \{ (x_1 \vee x_2 \vee x_3) \} \qquad G : \{ \neg (x_4 \vee x_5) \}$$

How would we classify some new instance x_6?

For any instance not in the training examples
half of the version space says +
the other half says –

=> To learn the target concept one would have to present *every* single instance in X as a training example (Rote learning)

Three Learners with Different Biases

- Rote learner: Store examples, classify x if and only if it matches a previously observed example.
  - No inductive bias
- Version space candidate elimination algorithm.
  - Bias: Hypothesis space contains target concept.
- Find-S
  - Bias: The hypothesis space contains the target concept & all instances are negative instances unless the opposite is entailed by other knowledge.

Summary

- Concept learning as search.
- General-to-Specific partial ordering of hypotheses
- Inductive learning algorithms can classify unseen examples only because of inductive bias
- An unbiased learner cannot make inductive leaps to classify unseen examples.